

# Entwurf und Implementierung eines Chrome Plugins zur Anreicherung von Webseiten mit kulturellen Inhalten

## Gliederung:

### 1 Einleitung und Motivation

#### 1.1 Unterschiede Web Augmentation, Web Personalization, Web Customization

→ WA to the web what augmented reality is to the physical world

#### 1.2 Wachsende Bedeutung von Just-in-time retrieval/ augmentation um dem Nutzer mehr Information zur Verfügung zu stellen.

##### 1.2.1 Veränderung des Suchverhaltens, mehr und bessere Informationen finden

##### 1.2.2 Vorteile/Nachteile gegenüber Suchmaschinen

- Vorteil Suchmaschine: Wenn der Benutzer eine klar Vorstellung von der Suche hat Suchmaschine besser. Oder er weiß genau wonach er sucht

- Vorteil JITIR: Man muss einen jetzigen Task nicht komplett unterbrechen → Man verliert nicht den Überblick was man gerade macht/bleibt im Kurzzeitgedächtnis

→ „JITIR agents greatly reduce the cost of searching for information“

→ Dafür sind Ergebnisse nicht so exakt wie bei Suchmaschinen

#### 1.3 Erleichtern von Recherchen durch einbinden von weiterführenden Links zu kulturellen Inhalten direkt in die betrachtete Webseite

#### 1.4 Verbesserung des EEXCESS Plugins/Unterschiede zum EEXCESS Plugin (Auswerten der alten Evaluierung?)

### 2 Related Work

#### 2.1 Unterschiede und Gemeinsamkeiten zu Just-in-time information retrieval agents (JITIR-Agent) (Rhodes)

##### 2.1.1 Proaktiv vs. User Interaction

##### 2.1.2 Informationen darstellen in „Nonintrusive Manner“

##### 2.1.3 Awareness of user's local context

#### 2.2 Vergleich mit Just-in-time information retrieval agents (Rhodes)

##### 2.2.1 Remembrance Agent (in EMACS Editor)

Zeigt Quellen an auf Basis des geschriebenen Texts, Benutzer kann Suchanfrage dann auch noch manuell anpassen/verfeinern

→ Vorteil von Suchmaschinen wird mit integriert

2.2.2 Margin? Web Plugin ähnlich wie EEXCESS

2.2.3 Watson

2.2.4 EEXCESS?

## 2.3 Text Retrieval Algorithmen

2.3.1 Term Frequency/Inverse Document Frequency algorithm

2.3.2 Text rank

## 2.4 Unterschiede und Gemeinsamkeiten zu „Automatic help systems“ (z.B. Microsoft Office Assistant → Domain spezifisch)

2.4.1 Domain spezifisch vs. Domain unabhängig

## 3 Konzept

### 3.1 Warum kein Proactiver JITIR-Agent?

→ API Limitierung und decrease cognitive load

→ Benutzer entscheidet ob er weitere Informationen erhalten möchte

→ Benutzer kann Suchanfrage erst anpassen (Nachteil von Margin Notes (Paper 4))

### 3.2 Anzeige der Ergebnisse

### 3.3 Erklärung der Such-Anfragen Generierung

### 3.4 Anpassen der Suchanfrage durch den Nutzer

### 3.5 Verbesserung der Suchanfrage z.B. durch maschinelles Lernen

## 4 Implementierung

### 4.1 Verwendung von AngularJS für alle Komponenten des Plugins

### 4.2 Bau der GUIs

→ Darf den Benutzer nicht zu sehr ablenken

→ Ergebnisse müssen in der Nähe ihrer „Quelle“ angezeigt werden (proximity compatibility principle)

→ Benutzer muss klar zwischen Webseite und Augmentation unterscheiden können

→ buntes, auffälliges Design

→ Ramping interface: Mehr Benutzerinteraktionen führen zu mehr angezeigten Informationen (Erklärung der Stages)

### 4.3 Einbindung der REST-Services

## 5 Evaluierung?

## 6 Future Work

### 6.1 Alternative Algorithmen zur Textanalyse

## 6.2 Implementierung einer automatischen Suchanfragen-Verbesserung durch maschinelles Lernen

6.2.1 Mehr kontextuelle Informationen miteinbeziehen

6.2.2 Such-Profil des Nutzers erstellen

## 6.3 Verbesserung der Ergebniss-Güte

6.3.1 durch Query Expansion

6.3.2 durch Filtern der Ergebnisse (mehr Präzision da Ausbeute bei JITIR nicht so relevant)

## 6.4 Anpassung der Anwendung auf mobile Nutzung

## 7 Conclusion

7.1 Steigerung der Effektivität und Produktivität von wissenschaftlichem Arbeiten

7.2 Starke Effektivitätssteigerung durch Punkte aus Future Work möglich?